The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 18 March 2003

The Patent Office

Patents Act 1977
Rule 16

Request for grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

1.	Your reference	GB920020076GB1
2.	Patent application number *(The Patent Office will fill in this part)*	0227250.8 22 NOV 2002
3.	Full name, address and postcode of the or of each applicant *(underline all surnames)*	INTERNATIONAL BUSINESS MACHINES CORPORATION Armonk New York 10504 United States of America
	Patents ADP number *(if you know it)*	5196370001
	If the applicant is a corporate body, give the country/state of its incorporation	State of New York United States of America
4.	Title of the invention	FAULT TRACING IN SYSTEMS WITH VIRTUALIZATION LAYERS
5.	Name of your agent *(if you have one)*	R J Burt
	"Address for Service" in the United Kingdom to which all correspondance should be sent *(including the postcode)*	IBM United Kingdom Limited Intellectual Property Department Hursley Park Winchester Hampshire S021 2JN
	Patents ADP number *(if you know it)*	7903925001

6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number	Country	Priority App No *(if you know it)*	Date of filing *(day/month/year)*
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date or the earlier application	No of earlier application		Date of filing *(day/month/year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.)

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	6
Claim(s)	3
Abstract	1
Drawing(s)	1

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	5
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application

Signature
R J Burt

21 November 2002
Date

12. Name and daytime telephone number of person to contact in the United Kingdom

P J Stretton
01962 815830

Patents Form 7/77

The Patent Office

7/77

Patents Act 1977
Rule 15

Statement of inventorship and of right to grant of a patent

The Patent Office
Concept House
Cardiff Road
Newport
South Wales NP10 8QQ

1.	Your reference	GB920020076GB1
2.	Patent application number *(if you know it)*	0227250.8 22 NOV 2002
3.	Full name of the or of each applicant	INTERNATIONAL BUSINESS MACHINES CORPORATION
4.	Title of invention	FAULT TRACING IN SYSTEMS WITH VIRTUALIZATION LAYERS
5.	State how the applicant(s) derived the right from the inventor(s) to be granted a patent	By employment and by agreement
6.	How many, if any, additional Patents Forms 7/77 are attached to this form?	
7.		I/We believe that the person(s) named over the page (and on any extra copies of this form) is/are the inventor(s) of the invention which the above patent application relates to. Signature R J Burt 21 November 2002 Date
8.	Name and daytime telephone number of person to contact in the United Kingdom	P J Stretton Tel: 01962 816057

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Peter DEACON
(UK resident)
c/o IBM United Kingdom Limited
Intellectual Property Law
Hursley Park
Winchester
Hampshire SO21 2JN
England

Patents ADP number *(if known)*
8511743001

Carlos Francisco FUENTE
(UK resident)
c/o IBM United Kingdom Limited
Intellectual Property Law
Hursley Park
Winchester
Hampshire SO21 2JN
England

Patents ADP number *(if known)*
7805781001

If there are more than three inventors, please write their names and addresses on the back of another Patents Form 7/77 and attach it to this form

REMINDER

Have you signed the form?

William James SCALES
(UK resident)
c/o IBM United Kingdom Limited
Intellectual Property Law
Hursley Park
Winchester
Hampshire SO21 2JN
England

Patents ADP number *(if known)*
8089112001

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Barry Douglas WHYTE
(UK resident)
c/o IBM United Kingdom Limited
Intellectual Property Law
Hursley Park
Winchester
Hampshire SO21 2JN
England

Patents ADP number *(if known)*

81670820001

Patents ADP number *(if known)*

If there are more than three inventors, please write their names and addresses on the back of another Patents Form 7/77 and attach it to this form

Patents ADP number *(if known)*

REMINDER

Have you signed the form?

FAULT TRACING IN SYSTEMS WITH VIRTUALIZATION LAYERS

Field of the Invention

This invention relates to error tracing, and particularly to error tracing in environments having virtualization layers between host applications and devices.

Background of the Invention

The problem of fault detection and isolation -- tracking down a problem in a complex system to its root cause -- is a very significant one. In some environments, there is simply a lack of any error reporting information, but in many enterprise-class environments, much effort is invested in raising and logging detected faults. In fault tolerant systems, such information is critical to ensuring continued fault tolerance. In the absence of effective fault detection and repair mechanisms, fault tolerant system will simply mask a problem until a further fault causes failure.

When a problem does arise, its impact is frequently hard to predict. For instance, in a storage controller subsystem, there are many components in the path or "stack" from disk drive to host application. It is difficult to relate actual detected and logged errors to the effect seen by an application or a user host system.

When many errors occur at the same time, it is particularly difficult to determine which of those errors led to a particular application failing. The brute force solution of fixing all reported errors might work, but a priority based scheme, fixing those errors that impacted the application that is most important to the business, would be more cost efficient, and would be of significant value to a system user.

Any lack of traceability also reduces the confidence that the right error has been fixed to solve any particular problem encountered by the user or the application.

Today's systems, with RAID arrays, advanced functions such as Flash Copy, and caches, already add considerable confusion to a top-down analysis (tracing a fault from application to component in system). It takes significant time and knowledge to select the root-cause error that has caused the fault.

With the introduction of virtualization layers in many systems, the problem is growing. Not only does virtualization add another layer of indirection, but many virtualization schemes allow dynamic movement of data in the underlying real subsystems, making it even more difficult to perform accurate fault tracing.

It is known, for example, from the teaching of United States patent number 5,974,544, to maintain logical defect lists at the RAID controller level in storage systems using redundant arrays of inexpensive disks. However, systems using plural such arrays together with other peripheral devices, and especially when they form part of a storage area network (SAN), introduce layers of software having features such as virtualization that make it more difficult to trace errors from their external manifestations to their root causes.

There is thus a need for a method, system or computer program that will alleviate this problem, and it is to be preferred that the problem is alleviated at the least cost to the customer in money, in processing resource and in time.

Summary of the Invention

The present invention accordingly provides, in a first aspect a method in a stacked system for associating errors detected at a user application interface of one or more of a plurality of host systems with root cause errors at a stack level below a virtualization layer comprising the steps of detecting an error at a user application interface; identifying an associated root cause error at a lower stack level; creating an error trace entry for said error; associating an error log identifier with said error trace entry; making said combined error log identifier and said error trace entry into an error identifier that is unique within said plurality of host systems in said stacked system; and communicating said error identifier to any requester of a service at a user application interface of one or more of a plurality of host systems when said service must be failed because of said root cause error.

Preferably, the step of making said combined error log identifier and said error trace entry into an error identifier that is unique within said plurality of host systems in said stacked system comprises combining an error trace entry and an error log identifier with an integer value to make an error identifier that is unique within said plurality of host systems.

Preferably, the root cause error at a lower stack level is in a peripheral device of said stacked system.

Preferably, the peripheral device is a storage device.

Preferably, the stacked system comprises a storage area network.

The present invention provides, in a second aspect, an apparatus for associating errors detected at a user application interface of one or more of a plurality of host systems with root cause errors at a stack level below a virtualization layer comprising: an error detector for detecting an error at a user application interface; a diagnostic component for identifying an associated root cause error at a lower stack level; a trace component for creating an error trace entry for said error; an identifying component for associating an error log identifier with said error trace entry; a system-wide identification component for making said combined error log identifier and said error trace entry into an error identifier that is unique within said plurality of host systems in said stacked system; and a communication component for communicating said error identifier to any requester of a service at a user application interface of one or more of a plurality of host systems when said service must be failed because of said root cause error.

Preferably, the system-wide identification component for making said combined error log identifier and said error trace entry into an error identifier that is unique within said plurality of host systems in said stacked system comprises: a component for combining an error trace entry and an error log identifier with an integer value to make an error identifier that is unique within said plurality of host systems.

Preferably, the root cause error at a lower stack level is in a peripheral device of said stacked system.

Preferably, the peripheral device is a storage device.

Preferably, the stacked system comprises a storage area network.

The present invention further provides, in a third aspect, a computer program product tangibly embodied in a storage medium to, when loaded into a computer system and executed, cause said computer system to associate errors detected at a user application interface of one or more of a plurality of host systems with root cause errors at a stack level below a virtualization layer, said computer program product comprising computer program code means for detecting an error at a user application interface; identifying an associated root cause error at a lower stack level; creating an error trace entry for said error; associating an error log identifier with said error trace entry; making said combined error log identifier and

said error trace entry into an error identifier that is unique within said plurality of host systems in said stacked system; and communicating said error identifier to any requester of a service at a user application interface of one or more of a plurality of host systems when said service must be failed because of said root cause error.

Preferred embodiments of the present invention for fault isolation in a virtualized storage subsystem in which errors are tagged with root cause information using unique error identifiers. This provides the advantage that multiple errors caused by a single fault in the system can quickly be diagnosed to the single fault. This speeds up the diagnostic procedure and reduces potential downtime in an otherwise highly available system.

Brief Description of the Drawings

A preferred embodiment of the present invention will now be described by way of example only, with reference to the accompanying drawings, in which:

Figure 1 shows an exemplary virtualization subsystem component stack.

Figure 2 shows an example of an error log according to a presently preferred embodiment of the invention.

Detailed Description of the Preferred Embodiment

The preferred embodiment of the present invention begins by taking the conventional error log (170), such as already exists in many enterprise-class environments. The error log is used to record faults that are detected by components in the system. These are typically the components that interface to the 'outside world', such as network or driver layers, that are the first to detect and then handle an error.

A unique identifier (210) is added to the existing, conventional, error log entry. This can be done by using a large (for example, 32-bit) integer for each entry. The unique identifier, when qualified by the identifier of the log, identifies a particular event that might subsequently cause I/O service, or other activity, to fail. The error log contains supplemental information detailing the fault detected (220), sufficient to allow a user or service personnel to repair the root-cause fault.

The unique identifier is then used as part of the response to any service request (for example, an I/O request) that must be failed because

of that error. The issuer of that request, on receipt of the failed response to its request, determines which, if any, of its own services or requests must be failed. It in turn fails its own requests, again citing the unique identifier that it initially received that identifies the cause of those failures.

Thus, the identity of the event causing failure is passed through the chain of failing requests, until it reaches the originator of each Request. The originator then has the information required to determine exactly which error event must be repaired for each detected failure, expediting the repair process, and ensuring that the most critical applications are restored first. Further, there is a higher degree of confidence that the correct error has been repaired, avoiding the time delay and associated cost of unsuccessful recoveries.

In the presently most preferred embodiment, the components that communicate the requests are layers in a software stack (100), performing functions such as managing RAID controllers (110), virtualization (120), flash copy (130), caching (140), remote copy (150), and interfacing to host systems (160). The method of the preferred embodiment of the present invention allows for traceability through the system down the stack to the edges of the storage controller.

Each component in the software stack may itself raise an error as a result of the original failing event. As an example, a write operation from an application service (190) may be returned as a failure to the SCSI back end (110), that is, the write was failed by the physical storage for some reason. This results in an error being logged and a unique identifier (210) being returned to the raising component. The failed write is returned to the layer above, along with the unique identifier. These are returned up to stack. At each layer this may result in a failure within that component - for example if a flash copy is active against the disk that failed the write the flash copy operation will be suspended and an error raised. This new error itself is assigned a unique identifier, but is marked with the unique identifier, or root cause (230), passed by the component below. The same may happen at each layer in the software stack. Eventually the initial error is returned as part of the SCSI sense data to the application server that requested the write.

The user can then relate the failed write operation down to the physical disk that failed the write, and the operations and functions that failed within the software stack - for example the flash copy operation described above.

It will be appreciated that the method described above will typically be carried out in software running on one or more processors (not shown), and that the software may be provided as a computer program element carried on any suitable data carrier (also not shown) such as a magnetic or optical computer disc. The channels for the transmission of data likewise may include storage media of all descriptions as well as signal carrying media, such as wired or wireless signal media.

The present invention may suitably be embodied as a computer program product for use with a computer system. Such an implementation may comprise a series of computer readable instructions either fixed on a tangible medium, such as a computer readable medium, for example, diskette, CD-ROM, ROM, or hard disk, or transmittable to a computer system, via a modem or other interface device, over either a tangible medium, including but not limited to optical or analogue communications lines, or intangibly using wireless techniques, including but not limited to microwave, infrared or other transmission techniques. The series of computer readable instructions embodies all or part of the functionality previously described herein.

Those skilled in the art will appreciate that such computer readable instructions can be written in a number of programming languages for use with many computer architectures or operating systems. Further, such instructions may be stored using any memory technology, present or future, including but not limited to, semiconductor, magnetic, or optical, or transmitted using any communications technology, present or future, including but not limited to optical, infrared, or microwave. It is contemplated that such a computer program product may be distributed as a removable medium with accompanying printed or electronic documentation, for example, shrink-wrapped software, pre-loaded with a computer system, for example, on a system ROM or fixed disk, or distributed from a server or electronic bulletin board over a network, for example, the Internet or World Wide Web.

It will be appreciated that various modifications to the embodiment described above will be apparent to a person of ordinary skill in the art.

CLAIMS

1. A method in a stacked system for associating errors detected at a user application interface of one or more of a plurality of host systems with root cause errors at a stack level below a virtualization layer comprising the steps of:

detecting an error at a user application interface;

identifying an associated root cause error at a lower stack level;

creating an error trace entry for said error;

associating an error log identifier with said error trace entry;

making said combined error log identifier and said error trace entry into an error identifier that is unique within said plurality of host systems in said stacked system; and

communicating said error identifier to any requester of a service at a user application interface of one or more of a plurality of host systems when said service must be failed because of said root cause error.

2. The method as claimed in claim 1, wherein the step of making said combined error log identifier and said error trace entry into an error identifier that is unique within said plurality of host systems in said stacked system comprises:

combining an error trace entry and an error log identifier with an integer value to make an error identifier that is unique within said plurality of host systems.

3. The method as claimed in claim 1, wherein the root cause error at a lower stack level is in a peripheral device of said stacked system.

4. The method as claimed in claim 3, wherein said peripheral device is a storage device.

5. The method as claimed in claim 1, wherein the stacked system comprises a storage area network.

6. An apparatus for associating errors detected at a user application interface of one or more of a plurality of host systems with root cause errors at a stack level below a virtualization layer comprising:

an error detector for detecting an error at a user application interface;

a diagnostic component for identifying an associated root cause error at a lower stack level;

a trace component for creating an error trace entry for said error;

an identifying component for associating an error log identifier with said error trace entry;

a system-wide identification component to make said combined error log identifier and said error trace entry into an error identifier that is unique within said plurality of host systems in said stacked system; and

a communication component for communicating said error identifier to any requester of a service at a user application interface of one or more of a plurality of host systems when said service must be failed because of said root cause error.

7. An apparatus as claimed in claim 6, wherein the system-wide identification component comprises:

a component for combining an error trace entry and an error log identifier with an integer value to make an error identifier that is unique within said plurality of host systems.

8. An apparatus as claimed in claim 6, wherein the root cause error at a lower stack level is in a peripheral device of said stacked system.

9. An apparatus as claimed in claim 6, wherein said peripheral device is a storage device.

10. An apparatus as claimed in claim 6, wherein the stacked system comprises a storage area network.

11. A computer program product tangibly embodied in a storage medium to, when loaded into a computer system and executed, cause said computer system to associate errors detected at a user application interface of one or more of a plurality of host systems with root cause errors at a stack level below a virtualization layer, said computer program product comprising computer program code means for:

detecting an error at a user application interface;

identifying an associated root cause error at a lower stack level;

creating an error trace entry for said error;

associating an error log identifier with said error trace entry;

making said combined error log identifier and said error trace entry into an error identifier that is unique within said plurality of host systems in said stacked system; and

communicating said error identifier to any requester of a service at a user application interface of one or more of a plurality of host systems when said service must be failed because of said root cause error.

ABSTRACT

FAULT TRACING IN SYSTEMS WITH VIRTUALIZATION LAYERS

In a stacked system, errors detected at a user application interface of one or more host systems are associated with root cause errors at a stack level below a virtualization layer by detecting an error at a user application interface; identifying an associated root cause error at a lower stack level; creating an error trace entry for the error; associating an error log identifier with the error trace entry; making the combined error log identifier and the error trace entry into an error identifier that is unique within the plurality of host systems in said stacked system; and communicating the error identifier to any requester of a service at a user application interface of one or more host systems when the service must be failed because of the root cause error.

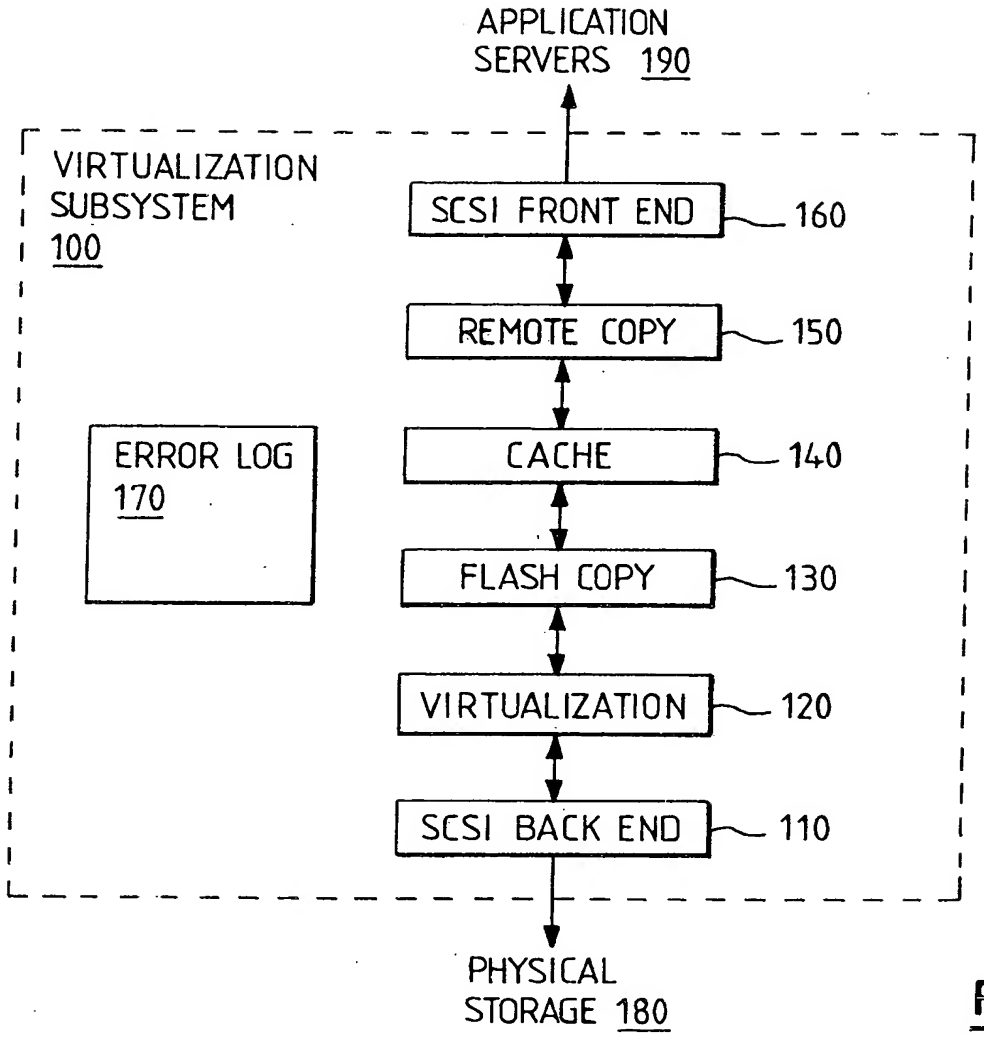

FIG. 1

ERROR LOG 170

SEQUENCE # 210	ERROR CODE 220	ROOTCAUSE 230
527	ABCDEF	
528	ABCDEE	527
529	ABCDED	527
⋮		
653	ABCCCC	527

FIG. 2